Faculty Working Papers

THE SEQUEL OF A FIRM'S INFORMATION PROCESSING WITH A DATABASE MANAGEMENT SYSTEM

Hirohide Hinomoto

#284

College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

FACULTY WORKING PAPERS

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

November 7, 1975

THE SEQUEL OF A FIRM'S INFORMATION PROCESSING WITH A DATABASE MANAGEMENT SYSTEM

Hirohide Hinomoto

#284

The Sequel of a Firm's Information Processing

With a Database Management System

Abstract

The organization in this study used all versions of IMS, from IMS 1 through IMS VS 1.01, over the period of 1970-1975. During this period, the number of messages processed steadily increased from 6,000 in October 1970 to a maximum in excess of 150,000 in September 1974. This volume enhancement was achieved by solving a sequence of problems concerning IMS software, database design, or program coding. Perhaps, the most important factors supporting the increase were two improvements in IMS software; one was the database buffer pool introduced with IMS 2, and the other the feature introduced with IMS VS that, unlike the previous versions, enabled the simultaneous updating of different database segments belonging to the same type.

Introduction

A great number of organizations have already converted or are currently planning to convert their computer systems operating in batch processing environment to computer systems with a database management system (DBMS) operating in on-line communications environment. In the past, absence of operational theories forced these organizations to acquire their skills in designing and operating the batch processing system through the tedious try-and-error method. Faced with the new processing environment, much of their hard earned skills has now become partially obsolete. No doubt, these organizations have to go through that tedious process of learning in this decade as did in the previous two decades, before developing an effective information system with a DBMS. This article discusses the case history of a firm that went through such a process.

The DBMS used by the firm in this study was Information Management System or simply IMS, a product of IBM. It has been one of the most advanced and widely used DBMSs commercially available and particularly popular among larger organizations. IMS was developed jointly by IBM and North American Rockwell Company in 1965 and released as a program product by IBM in 1968. It can accommodate both conventional batch processing and online message processing, either separately or concurrently. Some of the commercially available DBMSs have been considered superior to IMS in ease of use, response time and throughput under certain environments. IMS has been criticized by many because of its complexity imposing burden on programmers at user organizations. But it is also true that many large organ-

izations, including the firm reported here, seem to like IMS because of its flexibility.

To provide the reader with a brief explanation on how IMS (Version 2) processes a message, the following sequence of events and a schematic diagram in Figure 1 are given:

(1) The operator enters a message from the terminal keyboard and waits its output to be displayed on this terminal.

(2) IMS receives the message and writes information regarding the content of the message and the receiving time on the log tape.

(3) The message is placed in the input queue provided for its type.

(4) When it advances to the front of the queue and a region provided for its type becomes available, the message is removed from the queue and given a predetermined priority of processing. This is called scheduling.

(5) IMS obtains the database segments required by the message, places them in a message region, and processes them according to the message codes. If the message is an inquiry only, nothing is stored on the log tape, whereas if the message is an update, the contents of the database segments before and after the process are recorded on the log tape.

(6) The output is generated and placed in the output queue. At this time, its content is stored in the log tape.

(7) The secondary messages created by the terminal message are placed in the input queue in (3) above. From (3) to (7) will be repeated whenever a preceding message generates successor messages.

(8) Upon completion of the process of the current message, IMS records the time on the log tape.

(9) IMS places the output of the current message in the output queue and records the time on the log tape.

(10) The output placed in the queue in (6) or (9) is removed from the queue and sent to the display terminal.

(11) The terminal displays the output received.

The actual processing of a message is done as events (4) - (8) in a message region. Immediately before and after this process, the message resides in the control region under the control of its control program as input in events (2) - (3) or as output in event (9). During the period between event (2) and event (9), the message is subject to processing by the CPU. However, a terminal message does not necessarily go through the entire sequence of events in one continuous pass. It may generate a chain of background messages each of which goes through the cycle of events (3) - (7).

The firm studied here was a manufacturer of industrial electronic goods. It had work force of over 7000 and gross revenue exceeding 600 million dollars in 1974, the year of its peak business. Starting November 1974, however, the economic recession seriously affected the firm's business. By May 1975, the firm had laid off almost 40% of its work force. At this firm,

computer related activities were divided into three functional groups: the computer center, the system (programming) group and the application (programming) group. The system group was staffed by 10 system programmers including two specialists in IMS. The size of this group was not affected by the recession. However, this was not the case with the application group. In its peak in 1974, this group had 175 programmers of whom 50 engaged in IMS applications. By May 1975, its size was down to 100 programmers including 25 IMS programmers. The manager of the IMS application group also acted as the database administrator, which was the cause for some of the problems discussed later in this article.

Before the Installation of IMS

Early in 1969, the computer center of the firm was installed with an IBM 1400, an IBM 360/50, an GE (Honeywell) 415 and an GE (Honeywell) 430. The 1400 processed all types of business data in batch environment while the 360/50 processed both business and engineering jobs. In August 1969, the 1400 was replaced by an IBM 360/65. Subsequently, the existing 360/50 was used to emulate the 1400 while the new 360/65 processed all types of business and engineering jobs. All computers at this firm were leased from leasing companies for a minimum of six years and an average period of 7.5 years.

By this time, the firm had had some years of experience with time sharing through the GE DATANET 30 installed in the GE 430. The primary use of this system was the issue, tracing, and control of engineering documents through 9 to 12 CRT terminals. About 5,000 messages entering these

terminals per day were processed with data retrieved by an in-house access method. On the other hand, the GE 415 was used as an on-line monitor system to measure and record the results of product specification and reliability tests. These on-line systems gave the firm a sufficient background to deal with most problems in on-line realtime processing.

Early 1969, the firm initiated the development of an application program that would replace the aforementioned system processing engineering documents. Taking this opportunity, it was considering a broad scope improvement in its information processing operation. One of the urgent problems needing immediate attention was how to improve existing batch programs which were plagued with redundant processes. For example, one batch program processed the same file in seven different ways at seven different points in time when in fact these processes could have been combined into one. Further, information on most line items produced by the firm was typically stored in more than one file, partly because of the firm's multi-point warehousing and distribution system. Standard line items were kept at four warehouses, one for each of the four types of customers -- distributors, original equipment manufacturers, internal customers, and educational users. Separate inventory files were kept by the warehousing and shipping groups within each warehouse, which caused discrepancies in item volumes recorded in these files. In addition, different groups used different terms for the same item or followed different procedures for the same job, thus creating unceasing problems in communication and coordination. Further, where product lines were composed of components

produced by groups under different managers, the allocation of profit-and-loss figures to these managers was not clearly defined. To the information systems section, these organizational deficiencies were the causes for operational inefficiencies manifested by the excessive amount of time required for both the shipment of items available in inventory and the production and shipment of items not available in inventory. Further, ever increasing demands for the firm's products were aggravating the situation.

It was this general condition that led the information systems section to conceive an idea of entering data at different sources into common databases, and retrieving and updating them simultaneously by different operators in user departments. This meant the consolidation of all files into common databases, the independence of the databases from application programs, and a change in computer environment from batch processing to on-line real-time processing. The fermentation of these ideas took place towards the end of 1969. In these early years of the DBMS's history, currently well-known commercial DBMSs had yet to establish their reputation. From the beginning, senior members of the information systems section assumed that the firm's operation was too unique and complex to be satisfied with any of the existing DBMSs. As a result, no serious attempt was made to evaluate DBMSs then commercially available and select one for adoption. The firm was but one of many organizations in those years which tried to develop an in-house DBMS, but ultimately failed to complete the system.

In the early spring of 1970, two teams were organized. One team of eight programmers was to develop an application program to replace the existing program for engineering document issue and control run by the GE 430/DATANET 30. The new program was later called BOMP and its functions included the issue and control of engineering documents, issue and control of bills of materials, and routing of production orders. The other team composed of twelve programmers, was to develop an in-house DBMS that would be interphased first with BOMP and then with all future application programs for on-line messages. Both teams were to complete their projects by September 1, 1970.

The BOMP project progressed smoothly, but the DBMS project dragged its feet from the start. By the end of May, it became quite clear that the project could not be completed by the deadline. In June 1975, a team of senior programmers and systems analysts visited a West Coast firm which was said to have a successful use of IMS and came back with a report very favorable for IMS. These events finally led management to decide to abandon the development of the in-house DBMS and in its stead to adopt IMS. The decision was made in mid-July when the team was still laying out a master plan for the DBMS.

The primary reason for the project cancellation was economic. By the time the decision was made, management of the firm was aware of the fact that IBM had put a massive effort in the development of IMS and allocated huge resources for its future enhancement. To the management, committing a similar amount of resources in the development of the in-house DBMS was

clearly outside the firm's business. This feeling was strengthened by the minor economic recession that adversely affected the firm's financial position during that year. The decision to adopt IMS was strongly influenced by the opinions of those who had investigated the experience of the West Coast firm with the system.

Installation of IMS 1

Just before the installation of IMS in the IBM 360/65 in September 1970, the computer had 756 K bytes of core storage of which 600 K bytes were used for batch processing, leaving only 156 K bytes available for IMS. To enhance the capacity, one module of IBM's Large Core Storage (LCS) with one mega bytes and eight micro seconds in access time was added to main storage on a 30-day rental basis. In December, the IBM LCS was replaced by one unit of AMPEX's LCS with one mega bytes on a two year lease basis, because the latter was superior in performance (two micro second in access time) and cost only 1/3 as much as the former.

In its initial phase of installation during the period of September - December 1970, IMS Version 1, simply called IMS 1, failed to deliver the expected performance. The problem was mainly one of maintaining the software. IBM supplied the firm with a list of instructions to correct 90 known incidents caused by coding bugs. But the shortage in the IMS programming staff created by the redeployment of some members to the BOMP project made it impossible to complete the debugging until January 1971. Because of the successful shift of the work formerly done by the GE 430/ DATA NET 30 to the 360/65 with IMS, the former system was taken out in

February 1971.

The original version of BOMP was completed in October 1970 and became the first application to use IMS. But it went through a sequence of improvements continuously. Around January 1971, it processed some 6,000 terminal messages a day that entered six Sanders 620 CRT terminals. Most these messages were queries on engineering drawings, factory completion dates, components requirements, geographical location codes, and product codes. The response time was 5 to 7 seconds per call for about 98% of the terminal messages. But the remaining messages took a very long response time, three hours in several occasions and five hours in the worst case.

These exceptional cases appeared at random and independent of the load on the IMS at the moment. In these cases, terminal messages on line items created what appeared to be an endless chain of repeated explosions of product items at each of which an item was decomposed to sub-items for update. This was caused by the way the database was defined for the application; the database was divided into too many small segments which were placed at as many as five hierarchical levels. By August 1971, the redesign of the database was completed, finally solving the problem of response time. It included mainly reduction of the hierarchical levels into three levels and consolidation of small segments into a larger segment whenever feasible. The BOMP program itself was considered satisfactory and virtually unchanged since then. When the response time to a terminal message became very slow, the terminal operator would call the master operator at the computer center and requested an interruption and termina-

tion of the processing. Or if the terminal operator detected some errors in processing, she would request the master operator to reprocess the original message for there was no way for the master operator to determine if the processing of a message by IMS was taking too long.

<u>IMS2 replaces IMS1</u>

In October 1971, IMS Version 2, or simply IMS 2, replaced IMS Version 1. About the same time another module of AMPEX's LCS with one mega bytes was added to the 360/65's main storage primarily to increase the batch processing region. During the first few months of the installation, the new version of IMS was plagued with initial software bugs. A series of unpredictable problems buffled IMS system programmers. In one case, the malfunctioning IMS software stored data in such a manner that the standard recovery procedure became ineffective. The complexity of the system caused the debugging effort to last until the end of 1971.

IMS 2 was superior to IMS 1 in three main points. First, IMS 2 introduced two new access methods, the Hierarchical Indexed Direct Access Method and the Hierarchical Direct Access Method. Particularly, the former access method improved the efficiency of storage use by its capability of reusing locations vacated by deleted data. Such a capability was absent in the Hierarchical Indexed Sequential Access Method, the only indexed access method available to IMS 1.

Second, the introduction of a database buffer pool with IMS 2 greatly improved the response time by making active segments of the database readily available. A 'segment' is a group of fields retrieved as a unit. IMS

processes a message by fetching the necessary segments of the database into a region in main storage called a 'message region'. Upon completion of the processing, under IMS 1 the used segment is sent back to the database immediately. Under IMS 2, however, it is placed in an unused part of the database buffer pool, or if no unused part is big enough, it replaces the oldest segment in the buffer pool. When a message is scheduled for process, IMS first tries to find the necessary segment in the database buffer pool. If the segment is not found there, it is retrieved from the database. Database segments frequently used by messages have high probabilities of being found in the buffer pool and readily available for subsequent processing, thus saving the time required for search and transfer between disk storage and main storage.

The third improvement available with IMS 2 concerns the log tape provided for possible data recovery, one of the valuable features of IMS. With IMS 1, all database segments fetched by messages are recorded on the log tape regardless of whether there are changes in the segments, whereas with IMS 2 only those segments which have undergone changes are recorded. Consequently, IMS 2 might need only 4 or 5 tape reels to log messages when IMS 1 might have required 8 or 9 tape reels.

In October 1971, the Inventory Maintenance and Control Program (IMAC) with its initial module on inventory control of line items was installed as the second application to be processed by IMS. Subsequently late in November 1971, the Customer Order Processing Program (CORP) became the third application and shared a common database with IMAC. By the end of 1971, a maximum of 20,000 IMS messages were entered daily from about 30 Sanders

620 CRT terminals.

Meanwhile, the function of IMAC was expanded with additional modules. These modules exploded every ordered line item not available in inventory into piece parts requirements, and recorded the movement of every lot of goods for quality control, marking, packaging, or warehousing. When a lot was rejected in quality control, one of the modules initiated the reordering of a new lot. By early 1972, IMAC and CORP were very busy receiving a great number of terminal messages. In particular, messages for CORP caused a general deterioration in terminal response time.

Late in March 1972, an IBM 370/165 replaced the IBM 360/50. The new CPU was to process only IMS messages while the existing IBM 360/65 was run exclusively for batch and TSO jobs. But the 370/165 had to go through the initial shakedown common to new hardware, which further compounded the problem of deterioration in response time rather than improving it. By the end of May, the hardware problems were solved and the 370/165 processed IMS messages satisfactorily. Consequently, the 360/65 was finally taken out and the 370/165 started to handle both IMS messages and batch and TSO jobs. Meanwhile the total number of terminals installed for IMS messages had steadily increased from about 30 units of Sanders 620s at the end of 1971 to about 75 units in the middle of 1972. Of the 75 units, 60 were Sanders 620s used for the BOMP and CORP applications, and 15 were IBM 1050s and 2740s used as auditor terminals for exception reporting with IMAC.

During June 1972, IMS processed between 20,000 and 30,000 messages, of which 7,000 to 8,000 belonged to BOMP, about the same amount to IMAC, and the rest to CORP. The response time to a message was about 8 seconds on the average, 2 to 3 seconds during low load periods and a maximum of 15

to 20 seconds during peak load periods. The performance of IMS was considered very satisfactory.

Until the summer of 1972, each order entering CORP went through processes concerning customer characterization, preparation of order forms, shipping, and invoicing which were done by the following 8 separate modules: ORDER HEADER, ORDER SHIP-TO/SOLD-TO, MULTIPLE LINE ITEMS, ORDER LINE ITEMS, ORDER PRICE, ORDER INSTRUCTIONS, PRINT DITTOMASTER, and SHIPPING DATA/LINE ITEMS SHIPMENTS. During the summer, the CORP application was expanded to include 3 to 4 times the above number of modules. The previous modules processed only orders on line items, whereas the new ones not only processed orders on line items available in inventory but also exploded each line item not available in inventory into sub-items and then let IMAC to handle various processes on sub-items such as inventory control, production orders, material issue, and floor material control. Thus a single order entering through CORP initiated many background messages to be processed by IMAC.

The large volume of background messages issued by CORP created two problems; first, a long process time was required for each order entering CORP; and second, the prolonged processes initiated by CORP messages produced frequent intent conflicts with terminal messages for IMAC, because CORP and IMAC used a common database. An intent conflict in IMS 2 is a contention between messages trying to update segments belonging to the same segment type. For example, if two operators sent messages to update the inventory volumes of two different line items, that would create an intent conflict, because these volumes are stored in segments belonging to the same type.

In this situation, IMS 2 like IMS 1 allowed only the first message to update the segment concerned and let the remaining messages wait in the queue for their turns. Until the expansion of CORP in the summer of 1972, the intent conflict had never been a serious problem for the firm.

IMS 2.2 replaces IMS 2

In October 1972, IMS 2.2 replaced IMS 2 then in use. IBM never marketed IMS 2.1. The introduction of the new version was accompanied by initial bugs usually associated with new software. Efforts to eliminate these bugs continued through February 1973. Until this time, the expanded format of CORP introduced during the previous summer covered only a limited number of line items. With the introduction of IMS 2.2, the coverage was extended to the entire line items.

In the ensuing period, November 1972 - March 1973, the expanded coverage resulted in a great increase in the number of terminal messages related to CORP and IMAC, causing a serious deterioration in response time. During this period, the number of IMS messages reached 60,000 per day with an average response time of 45 seconds - 1.5 minutes and a peak load response time of 2 to 3 minutes around 11 a.m. and 2:00 - 3:30 p.m.

Around March 1973, a team composed of 2 system programmers and 4 application programmers was formed for a project to improve the serious deterioration in response time. This project lasting through May 1973 introduced changes in hardware and software. A major improvement in hardware was an addition of one mega bytes to main storage of which 200 K bytes were allocated to IMS. This brought about two improvements in the layout of IMS:

(1) an increase in the size of the database buffer pool from 30 K bytes to 50 K bytes, and (2) an increase in the number of message regions from three to four. Although both changes were believed to be responsible for a great improvement in response time, the first one was considered the primary factor. Terminal messages were now possible to find necessary segments of the database readily available in the expanded database buffer pool 75% to 80% of the time in contrast to 60% to 70% previously.

The change in software concerned application programs. It was aimed at reducing the probability of an intent conflict by reviewing two types of update codes available to IMS, straight UPDATE and mixed QUERY/UPDATE, both of which are equally potent in creating an intent conflict. Until the time of the improvement project, application programmers had generally been not fully aware of the seriousness of deterioration in response time caused by intent conflict and carelessly used QUERY/UPDATE even when the update did not always follow the query. Every existing message with QUERY/UPDATE was examined and broken up into QUERY and UPDATE whenever feasible. This modification was to improve the response time in two ways: (1) the query process could be performed free from an intent conflict, and (2) the lock-up duration of the segment type would be shorter with straight UPDATE than with mixed QUERY/UPDATE even if UPDATE had to follow QUERY.

The above modifications in hardware and application programs brought about a major improvement in response time. Before the modifications, around March 1973, intent conflicts caused 40 to 50% of the terminal messages to wait in the queue. In the worst case, as many as 30 messages were found in the queue. At that time the response time to a terminal message

was about 2 minutes in peak load periods. After the modifications, the response time was reduced to 3 to 5 seconds in normal periods and 8 to 9 seconds in peak load periods with at the most a few messages waiting in the queue at any time.

As a part of the improvement project, the firm invited an IMS consultant from outside to diagnose the CORP and IMAC programs whose messages constituted the major portion of the current IMS workload. Upon completion of his investigation, the consultant submitted the following recommendations:

(1) Two modules C41 and C98 of the order entry application process about 5000 messages each day. C25 enters orders on individual line items separately into the document database and then automatically generates the processes of C98. C98 in turn processes a number of things, such as checking demand details against inventory or releasing piece parts from the shop, many of which are also done by C41. Both C98 and C41 create three levels of processing for each exploding transaction and read the same series of segments at all three levels. The suggestion is the consolidation of C41 and C98 and the combination of the three levels into one for a major portion of the work.

(2) Module C61 of the customer order processing application processes the shipping transaction. For each line item ordered, it initiated a large number of background messages to be processed by module C62. This module releases piece parts from inventory

for the ordered item and prints out a usage detail for each of the released piece parts. For example, a single C61 transaction may generate up to 99 background messages each of which may subsequently generate 99 more. This explosion alone creates a significant load on the system. Each third level message may require to search exceedingly long overflow chains, some of which are as long as 6000 segments. In one case, nearly 200 overflow chains were required to process three third-level messages. The suggestion is to divide C62 messages into groups, each composed of some 10 messages, and to process these groups separately.

(3) No strong database administration function exists at this firm. Instead, this function has been left to the application group. Currently, a new independent database is created by every new application. The result is a proliferation of database. This is a type of problem that should be controlled by database administration. It is suggested to create a database administrator who must be independent of the programming groups and have the power to assure the company interest ahead of individual applications.

(4) System design review is another area requiring an immediate attention. It appears that new applications are not subject to any kind of technical review. Since each new group tends to use new personnel, the same mistakes are repeated many times. Benefits from previous experiences are not filtering down to system

designers and programmers.

The first two recommendations were acted on by the application programming group later in the year. But the last two recommendations regarding the absence of a database administrator in its true sense touched on the firm's internal organizational problems and have not been implemented at the time this report is written in May 1975.

In May 1973, the account receivable application (AREC) was installed as an extension of the order processing application (CORP). The new application used a database that had hitherto been used in batch environment. It processed messages for updating such data in the customer account as debits and credits, and shipments of ordered items. In addition, it determined the ages of account receivables and monitored the customer payment performance. The database used for AREC was balanced every night using all transactions on shipments and receivables stored on log tapes during the day. During the summer of 1973, AREC received 4,000 to 5,000 messages one half of which were processed on-line with an average response time of 5-8 seconds in non-peakload periods and about 30 seconds in peakload periods.

Meanwhile, the total volume of messages handled by IMS steadily increased from about 75,000 per day in March 1973 to about 95,000 in the following June, of which eight percent required updates. IMS was performing satisfactorily with only occasional complaints coming from user departments when the response time became exceptionally poor.

<u>IMS 2.3 replaces IMS 2.2</u>

Around the middle of July 1973, IMS 2.3 replaced IMS 2.2. The new

version was said to incorporate some features to decrease the CPU time required for overhead work and to reduce the probability of an intent conflict. About this time, some of the databases were redefined. The combination of the conversion to IMS 2.3 and the changes in the database produced a significant improvement in the IMS performance. Taking the same time frame in a peak period, 10 - 11 a.m., thruput on a typical day had increased from about 8,000 messages under IMS 2.2 to about 9,400 messages under IMS 2.3, thus showing an improvement of about 17%.

In the summer of 1973, the computer center had one computer, an IBM 370/165 with 3 mega bytes of main storage which were allocated to the following uses:

IMS	896 K bytes
TSO	154 K bytes
HASP	128 K bytes
OS/MVT	250 K bytes
TCAM	80 K bytes
LINKPACK	120 K bytes
Batch	the remainder

Of the 896 K bytes for IMS, 632 K bytes were allocated to the control region and 264 K bytes to 4 message regions. The allocation of storage in the control region was:

Data Base Buffer Pool	50 K bytes
Terminal I/O	30 K bytes
General Purpose Pool	30 K bytes
PSB Pool	40 K bytes
IMS software & control blocks	the remainder

Around this time, the main part of the peripheral equipment consisted of three IBM 3330 disk drives having a total of 24 spindles, each with 100 mega bytes, of which seven with a total of 700 mega bytes were used to store databases for IMS. At the end of June 1973, the channel associated with these units was busy 70 - 80% of the time, indicating a high probability of channel contention. Other peripheral equipment included 24 IBM 3420-5 magnetic tape drives, an IBM 2305 fixed head disk drive, and 135 terminals of various types. Two of the 24 IBM 3420-5 tape drives were used to maintain IMS log tapes, one of which was in operation at the time and automatically switched to the other as soon as it came to an end. The IBM 2305 fixed head disk drive was used to store Application Control Blocks (ACB), the operating system, and TSO message swapping. ACBs contain information necessary for defining the databases, data structures and access methods for application programs. Of the 135 terminals, 109 terminals were for IMS messages, including 48 units of IMAC, 15 units of BOMP, 8 units of CORP, 5 units for the account receivable application, 1 unit for the account payable applications, 26 units for special projects, and 6 units for use by system programmers and computer center staff.

Heretofore, the existing IBM 370/165 processed all types of jobs including those of IMS, TSO, RJE and batch. Since IMS messages were given the highest priority, the other types of jobs competed one another for the CPU capacity unused by the messages, which created considerable deteriorations in the performances of TSO and RJE. In particular, the response time of TSO was very unpredictable, fluctuating from one minute in one occasion to as much as 7 minutes in another occasion.

order to improve this situation, in August 1973, an IBM 370/158 with 1.5 M bytes of main memory was installed to operate under OS/VS and to relieve some burden from the overworking 370/165. The new CPU, rented during hours between 6 a.m. and 7 p.m., took over the entire jobs of TSO and RJE from the existing 370/165. TSO jobs were mostly those of production line scheduling, engineering design, and development of all types of programs, while RJE jobs were remote batch jobs mainly on business operation. To process TSO jobs, an IBM 2305 disk unit provided the 370/158 with six regions of virtual storage, each with 192 K bytes.

Although the 370/165 was dedicated to IMS messages during the day, it took over TSO and RJE jobs from the 370/158 after 7 p.m. when the latter CPU's rental period ended, and also processed batch jobs entered at the computer center during the third shift. The addition of the 370/158, however, did not significantly improve the performance of IMS, because IMS had already been given the highest priority to use the existing 370/165.

In October 1973, the IBM 370/165 was replaced by an more powerful IBM 370/168 with two mega bytes of main memory which was allocated to various software in the same manner as with the 370/165. The only difference was that the daytime batch region was reduced under the new 370/168.

The conversion from the 370/165 to the 370/168 meant a 20% increase in CPU power according to IBM experiments. In these experiments, IBM defined the CPU power by MIP representing millions of instructions processed per second and found the CPU powers of the 370/158, 370/165, and 370/168 to be .8 MIP, 2.0 MIP, and 2.4 MIP, respectively. These results were supported by the firm's experience in processing IMS messages. Namely,

on June 29, 1973, the 370/165 processed 93,900 IMS messages spending a total of 54 minutes and 41.7 seconds of CPU time or an average of .035 seconds per message, whereas on October 31, 1973, the 370/168 processed 79,300 IMS messages spending a total of 29 minutes and 8.1 seconds of CPU time, or an average of .023 seconds. Therefore, the change in CPU resulted in a 34% decrease in average processing time per IMS message.

Meanwhile, the total number of IMS messages, though fluctuating day to day, steadily increased as new modules were introduced. In August 1973, the incoming quality assurance testing module was added to the inventory maintenance and control application (IMAC). By the fall of 1973, the sustained increase in the number of IMS messages throughout the year caused a serious deterioration in response time. Consequently, starting that fall, the application programming group directed its main effort toward the tuning of existing application programs rather than the development of new ones. This provided the group an opportunity for the first time to implement the recommendations on applications made by the consultant in the previous spring. On the other hand, the system programming group thought that the tuning of existing applications might improve the IMS performance to some extent, but that a major improvement had to wait until the installation of IMS VS scheduled to be in the early part of 1974.

IMS VS 1.00 replaces IMS 2.3

In January 1974, the main memory capacity of the 370/168 was increased from 2 M bytes to 3 M bytes by the addition of an IBM add-on memory.

In March 1974 Message Formating Services (MFS) was installed in preparation for the forthcoming installation of IMS VS 1.00. It is a software package required by IMS VS for mapping input into output for IMS messages entered in IBM 3270-series terminals. Since only IBM 3270-series terminals could be interphased with IMS VS by MFS at this time, all Sanders terminals currently used had to be replaced by IBM 3277s.

Following the conversion of terminals, two futile attempts were made to install IMS VS simultaneously with OS/VS in the 370/168; the first attempt took place in April 1974, and the second May 1974. In both cases, the operating system stopped to function after working for 6 to 8 hours and sent out an IMS ABEND message because of a crush in data. The IMS emergency restart procedure was attempted but in vain. Had it been successful, it could have saved the current status of the system such as the existing queues, and the contents of the database buffer pool and message regions at the time of the crush. Subsequently, the IMS software was returned to IBM for debugging.

When a problem with IBM software is encountered as in the above case, IBM first examines whether it is formally registered in Retain 370 System, a system of two computers specifically to maintain a log of problems found in IBM software. If the encountered problem is logged in the system, this system dumps out all known events related to it. The dumped output is sent to the user for diagnosing and fixing the software at the user's premise. If the problem is not logged in the system, it goes through the second procedure. In this case, the software is sent to the IBM programmer who has written it. The programmer goes through the documentation, traps or dumps,

and fixes the bugs. Every time he fixes a problem, it becomes one of his test problems. When a new problem is encountered, it will be checked against those test problems.

The IMS VS software at this firm had to go through the second procedure. It was July 1 when the IMS VS software was returned for one more abortive trial on the following day. But, this time, the software problem was fixed during the same day. Thus, IMS VS 1.00 under OS/VS finally became operational on July 3, 1974, three months after its initial trial, formally replacing IMS 2.3 under OS/MVT. Since then IMS VS has been operating satisfactorily.

With the introduction of IMS VS, the Program Isolation (PI) option became available. Under this option, intent conflict would be created only if more than one message tried to update the same database segment, instead of the same segment type as with IMS 2.3. The improvement of IMS performance was confirmed indirectly by comparing the occupancy rates of message processing regions before and after the implementation of the option. One test conducted in October 1973, before the implementation of the option, showed that only two out of the five message regions were scheduled, whereas another test conducted in April 1975, after the implementation, showed that all of the 5 message regions were schedules. Both tests conducted during peak load periods when some messages were waiting in the queue.

The PI option and validity check incorporated in IMS VS were said to increased overhead in CPU time, but there was no way to measure their specific requirements for CPU time. Only data available for this type of analysis were those obtained by the DC Monitor, including the CPU time used

by each message in message region and the total CPU time spent by every 20,000 messages in control region. From the above data, an in-house program called the IMS Message Statistics obtained the actual total CPU time spent in message region and the estimated total CPU time spent in control region by messages belonging to each module. The estimation was done by allocating the total CPU time of 20,000 messages in control region to each module on the basis of the number of messages processed by the module, and the numbers of database calls and message calls made by these messages.

To compare the CPU requirements under IMS 2.3 with those under IMS VS, we have obtained average requirements per message in message and control regions for each of the groups of related modules, as is shown in Table 1. This table includes two sets of data; one set represents data on April 25, 1974, a date still under IMS 2.3, and the other set data on August 23, 1974, a date under IMS VS. Between the two dates, there was a substantial increase in CPU time in message region for all groups, whereas changes in CPU time in control region were by no means uniform among the groups. With regard to the overall averages for the total group , an average CPU time in message region increased 115% from .0268 seconds to .0568 seconds, while the average CPU time in control region increased only 8% from .144 second to .156 second. When overall changes are considered, the average total CPU time per message increased nearly 25% from .171 second to .213 second. Because of this substantial increase in CPU time, IMS VS did not deliver as much enhancement in thruput as the firm anticipated before its installation.

The stea'y increase in the numbe of messages with each message consuming pproximately 25% more CPU time under IMS VS gradually saturated the CPU capacity. This was confirmed by a histogram showing the hour-to-hour use of the 370/168 drawn from data obtained by the DC Monitor on a typical day in September 1974. The histogram showed that the CPU was busy almost 98% of the time during peak load periods.

As to the activities of the IMS application programming group, the purchasing and receiving modules added to IMAC in September 1974 were the only major modules developed between the summer 1973 and the end of 1974. They processed about 25,000 messages a week. Meanwhile, the group continued its effort for tuning the existing IMS application programs. During 1974, it spent approximately 60 man-months for program modifications and 20 man-months for message format modifications to improve the general response time to a terminal message.

One of the most important objectives of these activities was to increase thruput of terminal messages during peak load periods. This was to be achieved by restricting on-line processing to those messages which were to post or update data items needing an immediate attention. The remaining messages, including a minor part of the terminal messages and a major part, about 62%, of the background messages, were to be accumulated in disk storage for batch processing to be executed during the night. Up to this time, the background messages constituted on the average 51% of the entire IMS messages processed daily. They were initiated by terminal messages for the purposes of posting information, updating data in the data base, issuing orders, or executing tactical processes such as production scheduling

and control. Hitherto, these background messages were processed on-line with low priorities.

The most important in this respect was the elimination of module group D from on-line processing. The group processed on the average about 32% of the entire IMS messages or 62% of the background messages on a typical day; for example, 38,893 of the total of 121,967 messages on July 25, 1974, belonged to this group. The main part of this group was composed of messages concerning shipping and invoicing of ordered items, explosion of line items into piece parts requirements, and inventory control of line items and piece parts. Starting November 25, 1974, the elimination of module group D from on-line processing was in effect, radically decreasing the total number of IMS messages to be processed on-line (See Table 2).

As to changes in hardware, starting June 1974, IBM 3330 Model I disk units, with 100 mega bytes each, were gradually replaced by IBM 3330 Model II disk units, with 200 mega bytes each. Meanwhile, the number of IBM 3277 terminals handling IMS messages steadily increased and reached a maximum of about 300 units around September 1974. In November of the same year, the economic recession forced the firm to start laying off its employees. As a side effect, the number of terminals was reduced accordingly. As another anti-recession measure, the firm cancelled the scheduled installation of a new 370/168 which was to replace the existing 370/158 in November. Subsequently, in January 1975, the 370/158, a rented computer, was returned to IBM. Consequently, the existing 370/168 had to process all types of jobs besides IMS messages.

Effects of the recession were also apparent in the number of programming staff at this firm. The total number of application programmers was 175 before the recession, of which 50 worked on IMS applications. After November 1974, the number was reduced to 100 programmers including 25 working on IMS applications. On the other hand, the size of the system programming group was unaffected by the recession and stayed at 10 people, obviously to retain key talents essential to the maintenance of the computer system operation.

IMS VS 1.01 replaces IMS VS 1.00

In March 1975, IMS VS 1.01 replaced IMS VS 1.00, the original version of IMS VS. This was a preparation for the scheduled installation of OS/VS 3 in November 1975. But, no performance improvement was expected from the new version. Since the removal of the 370/158, the 370/168 as the only workhorse at this firm had been really taxed to its limit. As to its daytime use, TSO was given the highest priority to use 10% of its capacity and IMS the remaining capacity. A histogram shown in Figure 1 was constructed from data obtained by the DC Monitor on March 26, 1975, when a total of 64,893 IMS messages were processed. As the histogram indicates, the CPU was busy almost 97 to 98% of the time between 8 a.m. and 8 p.m. This was a typical situation since the start of the year.

In May 1975, the 370/168's main storage with 3 M bytes was allocated to the following purposes:

OS/VS	212 K bytes
Master Scheduler	128 K bytes

IMS	1216 K bytes
HASP	512 K bytes
LINKPACK	128 K bytes
TCAM	128 K bytes
TSO	128 K bytes
Others	the remainder

The 1216 K bytes allocated to IMS were used for the following purposes:

Control Blocks	141 K bytes
Data Base Buffer Pool	160 K bytes
Message Queue Pool	172.8 K bytes
Program Specification Blocks Pool	80 K bytes
Data Management Blocks Pool	56 K bytes
General Purpose Pool	24 K bytes
Terminal I/O	50 K bytes
Message Format Pool	120 K bytes
IMS Software	the remainder

Besides these, IMS was allocated with four message processing regions in virtual storage: three with 256 K bytes each, and one with 320 K bytes.

TSO jobs were processed in six different regions in virtual storage: (1) one region with 152 K bytes for 24 hours, (2) two regions with 256 K bytes each between 6 a.m. and 6 p.m., and (3) three regions with 192 K bytes each between 6 a.m. and 6 p.m. TSO jobs were divided into the following

four types:

(1) on-line development and execution of FORTRAN programs (80%)

(2) on-line development of COBOL programs (15%)

(3) on-line development of assembler programs mostly for system support (5%),

(4) occasional use for Programming Temporary Fixes by resident IBM programmers.

FORTRAN programs were mostly on engineering design and test, and a financial system estimating the profit or loss of a proposed venture, one of a few business applications using FORTRAN at the firm. All business applications including those related to the IMS were written in COBOL and debugged on-line through TSO. But, with the exception of IMS applications, all of them were run in batch environment.

Peripheral equipment installed at the firm as of May 1975 included:

(1) fixed head disk storage - two IBM 2305-2 disk drives and one IBM 2835-2 control units, which were used for paging and job queues in TSO and batch operations because of their fast access time,

(2) disk storage - 12 disk drives (six IBM 3330-1s, two 3330-2s, three 3333-2s and one 3333-1), each with two spindles, and three IBM 3830 control units,

(3) tape storage - 24 tape drives (16 IBM 3470s, four 3430-3s, and four 3450-5s) and three control units (two IBM 3800s and one 3803),

(4) printer systems - three IBM 3211-1 printers and three IBM 3811 print controllers.

(5) terminals - eight terminals for TSO (five IBM 2740s and three 2741s), 201 terminals for IMS (two IBM 3275s and 197 3277s at user departments, and one 2740 and one 3277 for IMS console), 32 printers (nine IBM 3284s and 23 3286s), 21 IBM remote controllers, and two 3272 local-to-local controller,

(6) System support - four terminals (one T.I. 733 and three T.I. 725s) and one ADM CRT, and

(7) others - several unit-record machines, and one Xerox S9-CPU time sharing system for engineering support and one RCA 70/35 computer for factory support.

In May 1975, the application programming group was still engaged in the improvement of the existing applications. Because of the cutback in programming personnel, the development of new applications were not planned for the coming months. The replacement of IBM 3330-1 disk units by IBM 3330-2 disk units with twice the storage density of the former in the previous summer created concentration of active segments on a few spindles. It was feared that there was a high probability of channel contention or even arm contention in the new environment. In particular, the product database was packed into one spindle when it was used by nearly 70% of the entire IMS terminal messages. To improve this situation, the redistribution of this database over a few disk spindles was currently under consideration.

Conclusion

Starting with IMS 1 in September 1970, the firm in this study used progressively every version of IMS as it was introduced by IBM. The number o IMS messages processed a day increased steadily from a modest figure of 6,000 around December 1970 - January 1971 to 20,000 in January 1972, 60,000 in January 1973, and 115,000 in January 1974, and finally reached its maximum exceeding 150,000 in September 1974. Table 2 lists the total number of messages and, where possible, the numbers of terminal and background messages processed on a typical day in each month since March 1973 through April 1975.

The rapid enhancement in messages processed daily at this firm was accomplished by alleviating the problem of serious deterioration in response time that accompanied the installation of every major IMS application. Frequently the solutions required modifications in database designs, message codes, or application programs. However, more important thrusts in thruput enhancement were brought about first by the replacement of IMS 1 by IMS 2 in October 1971, and then by the replacement of IMS 2.3 with IMS VS 1.00 in July 1974. Perhaps the most important new feature included in IMS 2 was the database buffer pool that radically improved the processing of a message requiring active database segments. On the other hand, the most important new feature included in IMS VS was its ability to simultaneously update different database segments of the same type, a process formally impossible under IMS 2.3.

The thruput enhancement by IMS VS 1.00 was much less than anticipated by the firm before the use of this version. Under the IMS 2.1 or 2.3, the

main barrier of thruput was intent conflict. IMS VS practically eliminated this problem, but was faced with the problem of an increased overhead in CPU time. With each message using about 25% more CPU time than before, messages under IMS VS saturated the available capacity of the IBM 370/168 in daytime operation by September 1974. The only possible way to increase thruput of terminal messages was to limit on-line processing to messages to post or update data segments needing immediate attention and store other messages in transition files for later batch processing. Starting November 1974, this was put into effect by modifying existing application programs. The result was a reduction of messages processed on-line by a little over 30%, alleviating the shortage in CPU capacity. Also contributing to the alleviation was a reduction in number of terminal messages because of the business recession.

Knowing that the business condition reached the bottom in May 1975, the manager of the computer center expected that the number of IMS messages would increase soon. Against this prediction, the first alternative he suggested was to add an IBM 370/158 to the existing 370/168 and bring back the same computing capacity that existed just before the recession. But his main concern was on the long run problem of how to deal with a volume of IMS messages exceeding the maximum that existed in the previous fall when the 370/168 was busy 98% of the time during peak periods of the day. He had a few alternatives in mind to deal with such a situation, but he considered all of them as marginal solutions in enhancing IMS thruput. What the firm would need within a few years, he thought, was some new technology for processing terminal messages with two to three times the thruput of the 370/168 with IMS VS.

To summarize, the firm's activities with IMS, Table 3 lists in chronological order various versions of IMS introduced, the computers and terminals installed, the average number of IMS messages processed daily the average and maximum response times, and IMS applications introduced since October 1970, when IMS 1 was first installed, through April 1975, in addition to the computers existed in 1969.

REFERENCES

1. IBM, Information Management System/360, Version 2, General Information Manual (GH20-0765-2), IBM, New York, 1972.

2. IBM, Information Management System/360, Version 2, System/Application Design Guide (SH20-0910-3), IBM, New York, 1972.

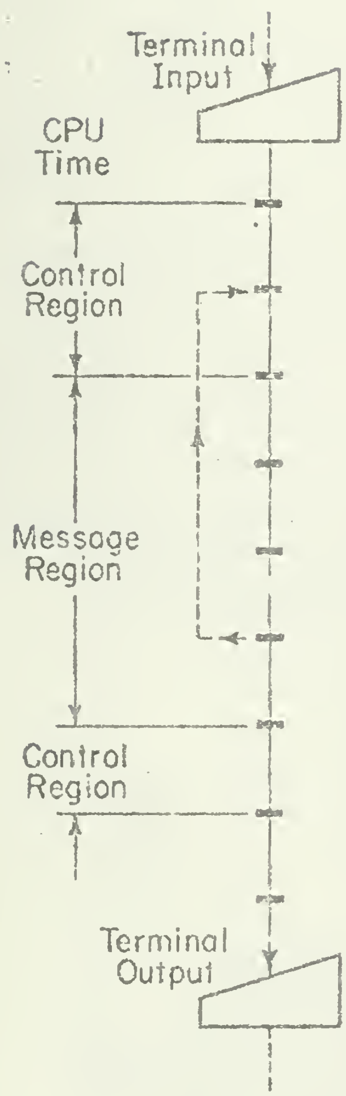
3. IBM, Information Management System/360, Version 2, Messages and Codes Reference Manual (SH20-0914-3), IBM, New York, 1972.

4. IBM, Information Management System/360, Version 2, System Programming Reference Manual (SH20-0911-3), IBM, New York, 1972.

5. IBM, Information Management System/Virtual Storage (IMS/VS), Conversion Planning Guide (GH20-9034-0), IBM, New York, 1973.

6. Cohen, L. J., Data Base Management System, A Critical and Comparative Analysis, Performance Development Corp., New Jersey, 1973.

Figure 1. Sequence of Events in Processing a Terminal Message with IMS.



Event

1. Enter a terminal transaction.
2. Receive the transaction and log the transaction content and receiving time.
3. Place the transaction in the input queue.
4. Remove the transaction from the queue for scheduling and log the time.
5. Process database segments: for an inquiry only, no log; for an update, log images before and after the process.
6. Generate output for terminal, put in the output queue and log the output.
7. Send back secondary transactions generated by the original transaction to the queue (Event 3).
8. Complete the process and log the time.
9. Send the output to the output queue and log the time.
10. Remove the output from the output queue.
11. Display the output on the terminal.

GP-70

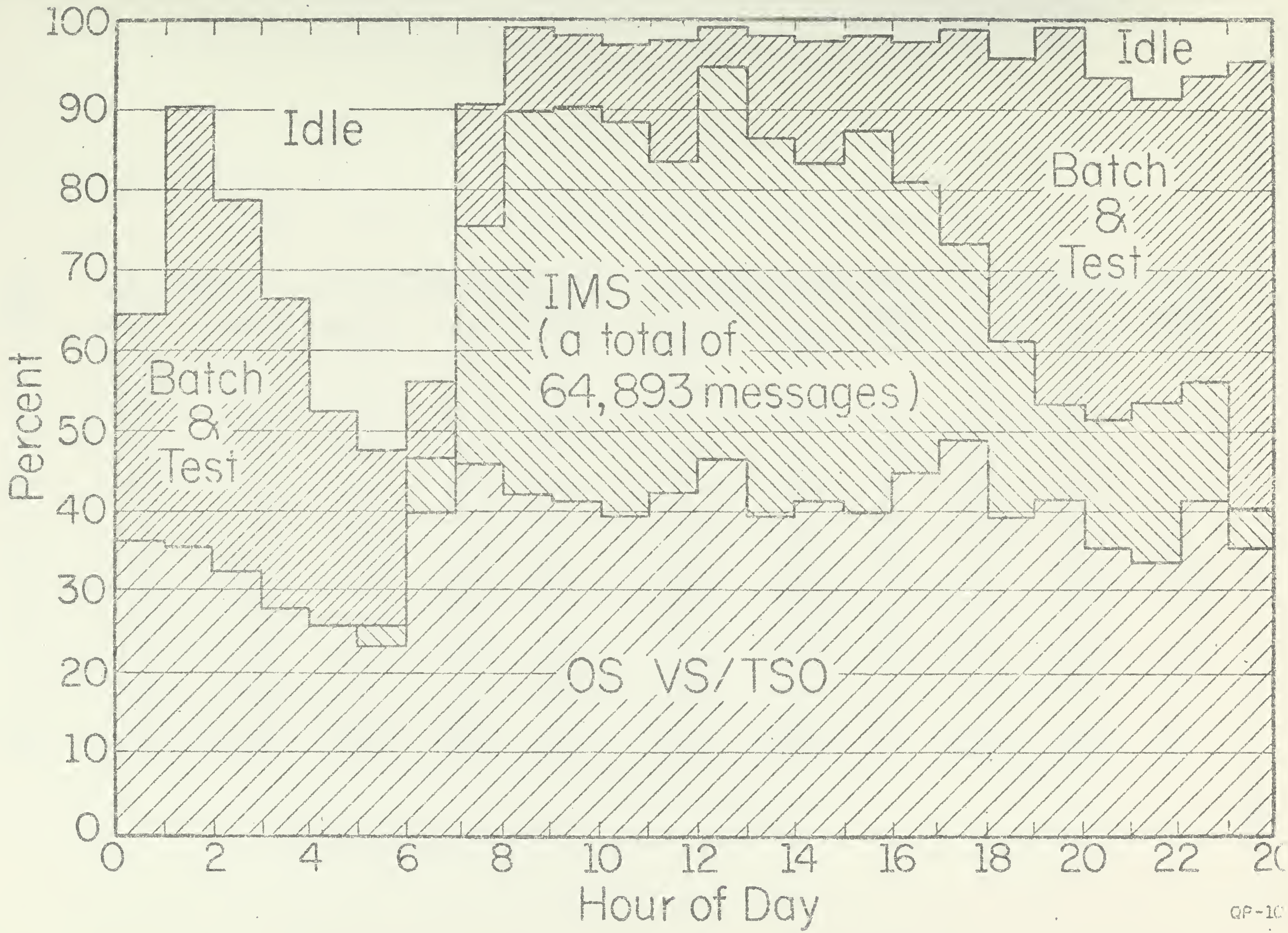


Figure 1. Hourly use of IBM 370/168 capacity on March 26, 1975

Table 1. Comparison of CPU Time Requirements on April 25, 1974, and August 23, 1974.

IMS Module Group	No. of Messages		Av. CPU Time in Message Region (1/1000 sec.)			Av. CPU Time in Control Region (1/1000 sec.)			Total CPU Time in Both Regions (1/1000 sec.)		
	April 25	August 23	April 25	August 23	Percent Change	April 25	August 23	Percent Change	April 25	August 23	Percent Change
A	1,427	1,967	29.45	58.91	100	67.27	70.30	4	96.72	129.21	34
B	27,147	30,729	32.27	69.65	116	249.62	260.71	4	281.89	330.36	17
C	35,337	37,977	7.29	16.44	126	48.17	43.09	-10	55.46	59.53	7
D	2,925	2,817	31.65	137.00	333	140.69	335.98	139	172.34	472.98	174
E	6,029	6,875	22.87	40.24	76	129.78	107.74	-17	152.65	147.98	-3
F	390	929	48.59	79.41	63	136.82	86.31	-37	185.41	165.72	-11
G	59	1,250	21.38	62.18	191	67.42	225.88	235	88.80	288.06	224
H	25,826	25,730	29.52	70.26	138	211.45	210.09	-1	240.97	280.35	16
I	3,167	3,496	71.29	128.06	80	141.55	131.33	-7	212.84	259.39	22
J	221	233	37.68	73.78	96	119.61	97.33	-19	157.29	171.11	9
K	3,492	1,720	153.38	324.16	111	220.01	271.31	23	373.39	595.47	59
L	8,421	3,736	13.07	39.95	206	32.56	57.68	77	45.63	97.63	114
M	64	106	11.03	18.36	66	16.14	15.88	-2	27.17	34.24	26
Total	114,505	117,560	26.80	56.78	112	144.37	156.57	8	171.17	213.35	25

TABLE 2. DAILY IMS MESSAGES SINCE MARCH 1973

DATE	TOTAL IMS MESSAGES	TERMINAL MESSAGES NUMBER	TERMINAL MESSAGES FRACTION OF TOTAL	BACKGROUND MESSAGES NUMBER	BACKGROUND MESSAGES FRACTION OF TOTAL	MODULE GROUP D MESSAGES NUMBER	MODULE GROUP D MESSAGES FRACTION OF TOTAL
3-30-73	79,670						
4-25-73	88,072						
5-25-73	102,891						
6-29-73	93,930	43,457	.46	49,973	.53		
7-27-73	88,868						
8-16-73	83,437						
9-28-73	95,060						
10-25-73	97,542						
11-21-73	109,300	55,347	.51	53,953	.49		
12-27-73	109,943						
1-23-74	118,000	61,913	.52	56,087	.48		
2-27-74	108,900	54,602	.50	54,298	.50		
3-27-74	93,200	42,939	.46	50,261	.54		
4-25-74	114,522	53,385	.47	61,137	.53	35,337	.30
5-22-74	110,200	48,194	.44	62,106	.56		
6-26-74	114,700	55,676	.48	59,024	.52		
7-25-74	121,967					38,893	.32
8-23-74	117,571					37,977	.32
9-26-74	153,040					50,146	.33
10-25-74	127,537						
11-21-74	134,682					45,122	.34
11-26-74	89,888					0	
12-10-74	78,318					0	
12-30-74	67,589					0	
1-24-75	77,118					0	
2-20-75	74,173					0	
3-25-75	68,240					0	
4-11-75	53,870					0	

Table 3. Main Chronological Events in Information Processing with IMS

Date of Event	New Version of IMS	New Set of CPUs Installed at Computer Center (* shows the CPU with IMS)	No. of IMS Terminals	Average Daily Messages	Response Time	New IMS Application (See Footnotes for explanations)
August 1969	In-House Access Method	1. IBM 360/65 2. IBM 360/50 3. GE 415 4. GE 430/DATANET30	9 units	5,000		Engineering Document Issue and Control
September 1970	IMS 1					
October 1970						BOMP
January 1971			6 Sanders 620s	6,000	av. 5-7 sec. a very long time for a few	
February 1971		1. IBM 360/65* 2. IBM 360/50				
October 1971	IMS 2					IMAC
November 1971						CORP
December 1971			30 Sanders 620s	20,000		
March 1972		1. IBM 370/165* 2. IBM 360/65				
May 1972		1. IBM 370/165*				
June 1972			75 terminals: 1. 60 Sanders 620s 2. 12 IBM 1050s 3. 3 IBM 2740s	30,000	av. 8 sec. max. 15-20 sec.	—
October 1972	IMS 2.2					

January 1973				60,000	av. 45-90 sec. max. 2-3 min.	
May 1973				90,000	av. 3-5 sec. max. 8-9 sec.	AREC
June 1973			109 terminals: 1. 55 Sanders 620s 2. 1 IBM 3275 3. 29 IBM 3277s 4. 24 Trivex 4488s	95,000		
July 1973	IMS 2.3			90,000		
August 1973		1. IBM 370/165* 2. IBM 370/158		95,000		Incoming Quality Assurance
October 1973		1. IBM 370/168* 2. IBM 370/158		100,000		
July 1974	IMS VS 1.00			120,000		
September 1974			Approx. 300 terminals	150,000		Purchasing and Receiving
January 1975		1. IBM 370/168*		77,000		
March 1975	IMS VS 1.01			70,000		
May 1975			201 terminals 1. 2 IBM 3275s 2. 198 IBM 3277s 3. 1 IBM 2740	40,000	av. 22 sec. max. 1.5 min.	

Notes: Explanations of Applications

1. BOMP processes the issue and control of engineering documents, issue and control of bills of materials, and routing of production orders.
2. IMAC is an application to ship ordered items, control inventories of line items, issue replenishment orders, and issue and route piece-parts production orders.
3. CORP processes customer sales orders and related activities.
4. AREC processes the updating of customer accounts, shipments of ordered items, determination of ages of account receivables, and customer payment performance.